SwipeBite - Bi-Weekly Progress

SungJun Park
Ramon Arias
Shakhzod
Yujin Juhn

Addressing Feedback

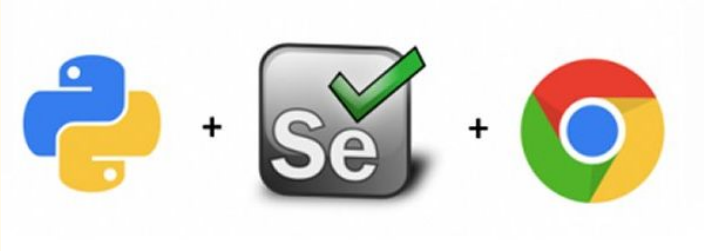
- Recommendation system needs a lot of data. How to get restaurant data? More specifically, there are few comments and review data and also limited.
- Considering the time and cost it takes, what is the big difference from the existing app(Kakao map, Naver).

- I think there are many options / filters, How can you make intuitive UI by using those filters?
- where do you get data?

In addressing the parking problem, could you please clarify how you intend to gather data and furnish information regarding the availability of vacant parking spaces, including their specific locations? It seems we might require additional cameras for this purpose.

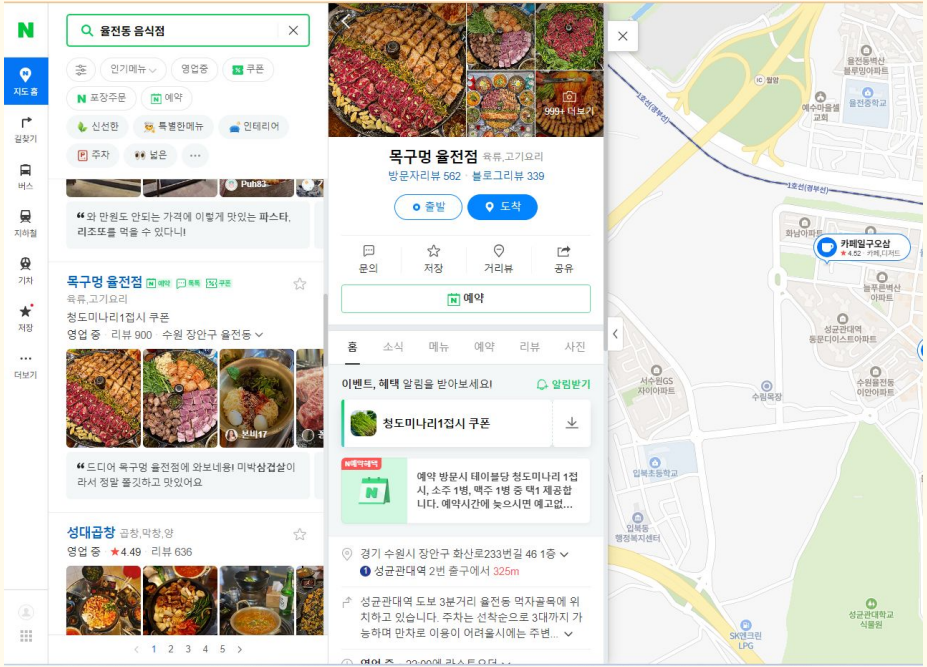
- **Data Collection:** Initially thought about using Google API or Naver API but now we are going to develop our own web crawler for Naver Maps.
- **Parking Information:** Our train of thought with this feature was to let the user know if there is parking on the premises of the restaurant. We are ***NOT*** providing whether these parking spaces are occupied or empty. We can get this from Naver Maps.
- **Too Many Filters:** We try to include a certain amount of filters so that it is quicker for the user to decide where to eat, an attempt to narrow down options.

Naver Maps Crawler



- **HTML Parsing:** We'll use HTML parsing libraries to extract the desired data from the page's structure.
- **Data Storage:** Extracted data will be stored in a structured format, such as a database or CSV files, for further analysis and use.

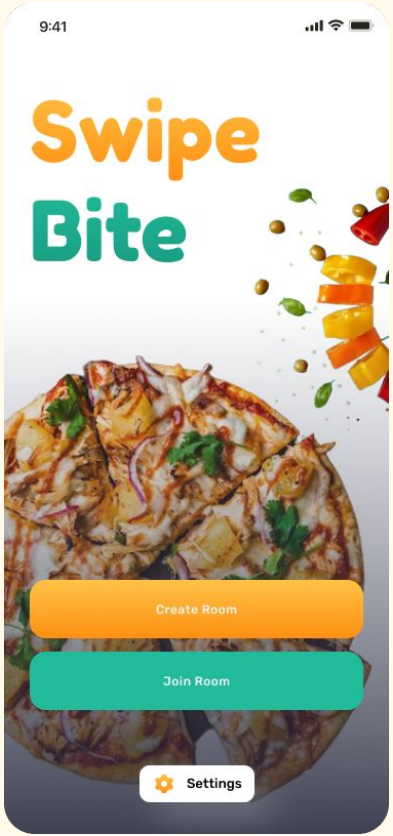
Naver Maps Crawler

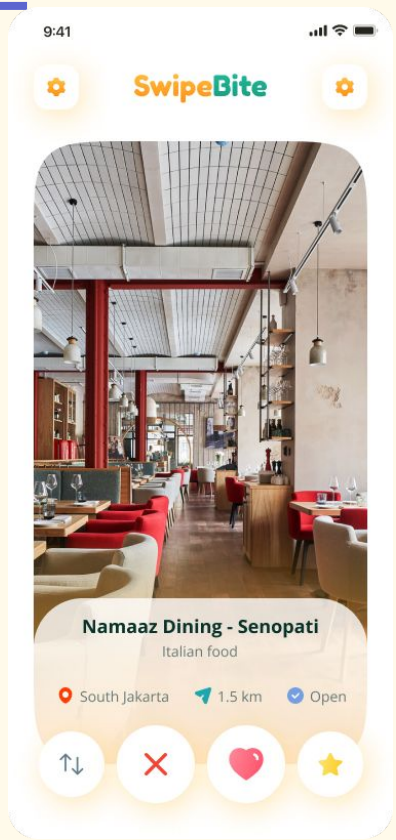


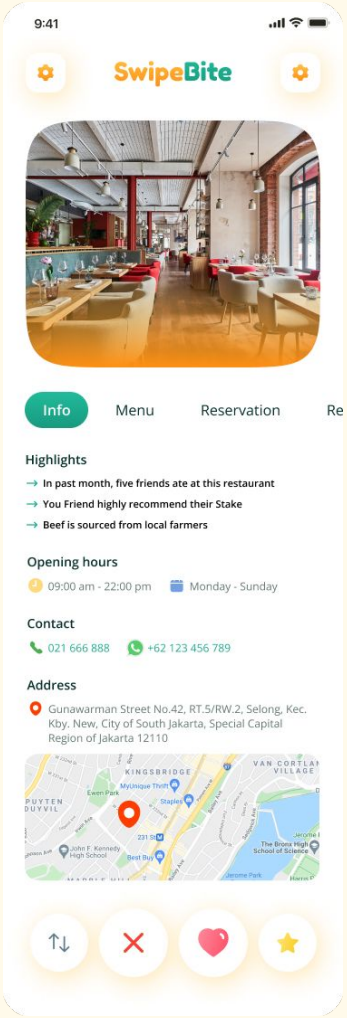
We managed to crawl

- Name of restaurant
- Category of restaurant
- Ratings
- Number of Reviews
- Several Photos
- Opening Hours
- Whether Parking is available
- Phone Number
- Menu Plate
- Dishes and Price

Initial UI Design







Project Plan - Role Assignment

- **Shakhzod** - Backend Team
  - Server - setting up infrastructure
  - Database - schema for storing profiles, food data, etc
  - Work on developing API
- **Sung Jun Park** - Backend Team
  - Continue developing crawler
  - Design and implement features for the app
  - Work on developing API
- **Ramon Arias** - Frontend Team
  - Design UI/UX
  - Implement design into functional code
  - Work on developing API
- **Yujin Juhn** - Backend Team
  - Continue developing crawler
  - Post-process on achieved data
  - Work on developing API